SPARTA: Security & Privacy Architecture through Risk-driven Threat Assessment

Laurens Sion, Dimitri Van Landuyt, Koen Yskout, Wouter Joosen
imec-DistriNet, KU Leuven
Heverlee, Belgium
{laurens.sion, dimitri.vanlanduyt, koen.yskout, wouter.joosen}@cs.kuleuven.be

***Abstract*—The development of secure and privacy-preserving software systems entails the continuous consideration of the security and privacy aspects of the system under development.**

While contemporary software development practices do support such a continuous approach towards software development, existing threat modeling activities are commonly executed as single-shot efforts leading to a single, historic, and quickly obsolete view on the security and privacy of the system. This disconnect leads to undetected new issues and wasted efforts on already resolved problems, effectively accruing technical debt.

The presented SPARTA prototype facilitates the consideration of security and privacy by providing support for: (i) capturing security and privacy design decisions in a DFD-based architectural abstraction, (ii) continuous threat elicitation on this knowledge-enriched abstraction, and (iii) risk analysis of the elicited threats for prioritizing security and privacy efforts. By capturing and continuously assessing the impact of security and privacy design decisions on the elicited threats, the progress towards securing the system can be assessed and alternatives can be compared, taking into account past and present design decisions.

***Index Terms*—Security, Privacy, Threat modeling, Risk analysis, Secure design**

I. Introduction

Security and Privacy by Design (SbD/PbD) are increasingly being recognized as essential principles for preventing the introduction of design flaws that compromise security or privacy [1]. Recently introduced laws, such as the EU General Data Protection Regulation (GDPR) [2], even mandate adhering to privacy and data protection by design, and by default, for all systems involved in the processing of personal data. A way to realize these principles in practice is the application of a threat modeling approach, such as STRIDE [3], [4] for security or LINDDUN [5], [6] for privacy. These approaches start from a Data Flow Diagram (DFD) [7] abstraction. They enable a rigorous and methodical security and privacy analysis of the system under design, by systematically iterating over the model elements to identify all potential security or privacy threats.

Afterwards, the discovered threats must still be manually assessed with respect to their importance (for example, based on likelihood and impact). This assessment ideally involves: (i) the consideration of existing countermeasures, (ii) other stakeholders, and (iii) the application of risk assessment methodologies [8], [9]. This to assist in the trade-off between introducing additional countermeasures (including their effectiveness) and their associated cost and impact.

However, even with the application of well-established risk analysis methodologies, performing this analysis in a separate activity requires considerable effort. Furthermore, it scatters the relevant knowledge across multiple artifacts, hindering the easy reuse of this knowledge when revisiting earlier decisions.

In this paper, we present the prototype of our SPARTA tool, implementing the earlier presented approach [10], which combines threat modeling and risk analysis in a single context. SPARTA supports the enrichment of existing threat modeling artifacts with the information necessary for performing a risk analysis. SPARTA can incorporate both these activities in a single analysis step. Similar to other tools [11], this step results in a list of threats. A distinguishing feature of SPARTA is that the resulting list of threats if further enriched with a risk estimate, based on Monte Carlo simulations that take into account the value of threatened assets, as well as the difficulty of various types of attackers to overcome the security and privacy countermeasures that are in place.

SPARTA provides several benefits to its users: (i) guidance in prioritizing security and privacy efforts towards the most important threats; (ii) including security and privacy countermeasures in the model and taking their effect into account during the threat elicitation and risk analysis; (iii) replacing an all-or-nothing 'mitigates' relation between a countermeasure and a threat with one that takes the countermeasure's strength and attacker's capability into account; and (iv) enable monitoring of the overall risk reduction progress with the introduction of countermeasures and keeping track of the residual risk.

This paper is structured as follows. Section II positions our work with respect to existing tools and approaches. Section III elaborates on SPARTA's design, implementation and usage. Next, Section IV discusses both initial and future evaluations. Section V lists some ideas for future extensions to SPARTA. Finally, Section VI summarizes our main contributions.

II. Related work

Threat modeling was introduced by Microsoft as part of its security development lifecycle [3], [4], [12]–[14] and has proven popular since with multiple real-world applications in industry [13]–[15]. Microsoft also provides tool support with its Threat Modeling Tool [11] and more recently, the OWASP project also provides the Threat Dragon tool [16], although it currently does not yet automatically elicit threats.

In existing approaches, DFDs remain largely security- and privacy-agnostic, with minor, often ad-hoc, additions for security or privacy. Recently, however, there are several proposals for a more systematic representation of security and privacy knowledge to elicit more relevant treats [17]–[19].

Complementary to threat modeling, risk analysis can be used [20]. Approaches such as CORAS [8] elicit security requirements starting from security goals or anti-goals. There are also tools in this space, such as Irius Risk [21], which focuses on the used technological components and associated risk instead of starting from the design of the application.

Finally, tools such as ThreatSpec [22] start from code-level enrichments to generate the DFDs for subsequent analysis.

An issue previously identified by Türpe [23] is the lack of addressing the interplay between the design, threat, and goal dimensions of security. This issue remains overlooked in the currently available tool support.

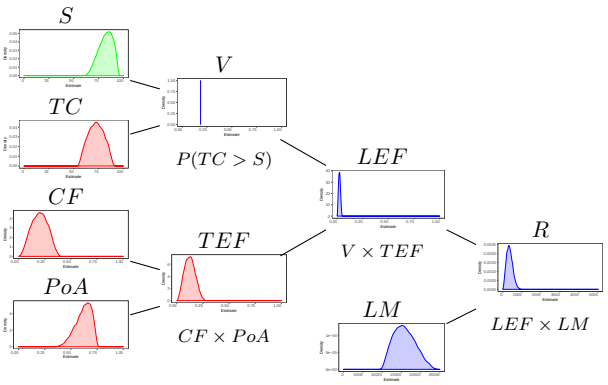

Fig. 1. Overview of the FAIR risk components and their combinations
The risk components from FAIR [9] are used to estimate the risk using Monte Carlo simulations. The nodes show example estimate distributions (graphs) and the lines illustrate which components are combined.
Legend: S: Strength, TC: ThreatCapability, CF: Contact Frequency, PoA: Probability of Action, V: Vulnerability, TEF: Threat Event Frequency, LEF: Loss Event Frequency, LM: Loss Magnitude, R: Risk

III. SPARTA

This section elaborates on the SPARTA prototype.[1] First, a high-level overview of the design and implementation are provided. Next, the practical usage of SPARTA is discussed.

A. Design & Implementation

At its core, SPARTA is based on typical DFD models that are used for threat modeling, extending them with security and privacy solutions and risk analysis information.

Plain DFD model: Basic threat modeling starts from a plain DFD model, which defines an abstract view of the system under design using four types of elements: process, data store, external entity, and data flow. For threat modeling, a fifth element type is added, namely trust boundaries. This model is subsequently used to elicit threats by matching model elements to certain predefined threat expressions. For example, information disclosure threats on a data flow that ends in a data store can be found by matching the expression '`*−flow−data store`'. Each threat type has such an expression to specify which combination of DFD model elements is vulnerable to that type of threat. By systematically matching the threat expressions to all elements, the list of threats is generated. This corresponds to the basic operation of the STRIDE [4] and LINDDUN [5] methodologies.

Enrichment with security and privacy solutions: The approach outlined above is completely unaware of any existing security or privacy countermeasures. Tools such as the Microsoft Threat Modeling Tool [11] take these into account to a limited degree by attaching simple properties to individual elements. The values of these properties can be used to exclude inapplicable threats during generation. For example, each data flow has a property *Provides confidentiality* which can be set to 'yes' or 'no'. When set to 'yes', information disclosure threats are no longer generated for that flow. While useful, such properties are very local, and their expressiveness is limited.

SPARTA supports a more extensive representation [19] of security solutions in the form of architectural patterns for security [24] and privacy. The patterns can express complex solutions that involve multiple DFD model elements (instead of a single element with the property). They define various roles that need to be bound to DFD elements, as well as a *Countermeasure* element that specifies which threat(s) on which element(s) are actually mitigated by the solution. By specifying the countered threats in the solution, the set of solutions can be extended without modifying the threat expressions.

Consider, for example, TLS for protecting against information disclosure and tampering of a data flow, and against spoofing of the server. Authentication in this context typically happens only in one direction (i.e., the client authenticates the server) and only for those flows that are part of the TLS communication (i.e., other flows to the server are not protected). A single property on the data flow or server process would be insufficient to capture this security solution correctly [19].

Risk analysis enrichments: Risk analysis explicitly takes into account uncertainty. SPARTA realizes this by implementing FAIR risk analysis [9], inspired by Bedra [25] who applied the Monte Carlo simulations for estimating the *Vulnerability*. However, SPARTA widens this approach by performing Monte Carlo simulations for each risk component from FAIR [9].

To enable this analysis, the underlying DFD model is extended with risk analysis information, in the form of estimates. Using a risk analysis methodology enables the assessment a threat's applicability on a continuous scale (namely its risk) instead of a binary scale (i.e., applicable or not applicable). This type of relation is much more realistic, as no security/privacy mechanism is perfect; in SPARTA, a partial reduction of a threat's applicability can still be taken into consideration.

Figure 1 shows the risk components from FAIR [9] that SPARTA uses. For each type of threat, the leaves in this tree (i.e., countermeasure strength, attacker capability, probability of action, contact frequency, and loss magnitude) have to be entered by security experts and other stakeholders. To handle the uncertainty on this information, each element

[1]More information at: https://distrinet.cs.kuleuven.be/software/sparta

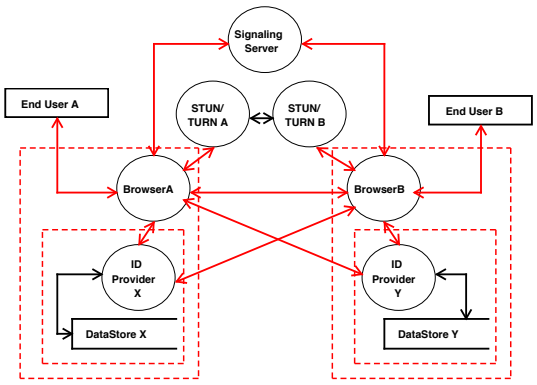

Fig. 2. WebRTC Data Flow Diagram used in the evaluation. Individual dataflow labels are omitted for readability.

is represented as an *Estimate* with four parameters: {min, probable, max, confidence}. One estimate defines a modified PERT distribution [26]. The leaves in Figure 1 contain examples of these distributions and shows how they are combined.

A risk estimation is performed for every threat that is generated using the threat generation expressions. To determine the risk for a threat, a Monte Carlo simulation is performed by sampling from the respective distributions, and combining the samples as shown in Figure 1. We highlight that the computation of a vulnerability converges to a single number, namely the probability that an attacker with the given capability distribution defeats a countermeasure with the given strength distribution. If multiple countermeasures are present (as part of a defense in depth strategy), the vulnerability becomes the probability of defeating all of them. Also note that the risk analysis always happens in the context of a specific attacker type, as a script kiddie has a different capability than a nation state to overcome certain security and privacy countermeasures.

B. Usage

This section briefly discusses the usage of SPARTA. Analogously to existing threat modeling activities, using SPARTA starts with the creation of a data flow diagram, such as the DFD in Figure 2, of the system under design. After creating the diagram, individual DFD elements are enriched with value *estimates*. These estimates specify the potential loss when a threat occurs. A single value is sufficient for the analysis, but SPARTA also supports specifying different estimates per threat type, for example when the disclosure of information would result in higher losses than unauthorized modifications.

In addition to the estimates, security and privacy solutions can be added as well. SPARTA imports external solution catalogs to enable reuse of existing security and privacy solutions. A solution type in a catalog contains the details on specific countermeasures (e.g., the TLS solution contains various encryption, integrity, and authentication countermeasures) and their strength, which are used in the risk analysis later on. Using a central catalog prevents the need for re-entering this information every time a solution is instantiated.

With that, all necessary information in the model is provided. Next, SPARTA's risk analysis is performed in the context of a specific attacker model. This attacker model specifies the *ThreatCapability*, *ProbabilityOfAction*, and *ContactFrequency* risk components, as illustrated in Figure 1. In principle, the threat modeler can specify any custom attacker type. To limit the required inputs, SPARTA offers a built-in set of attacker types, ranging from an opportunistic attacker to more advanced attackers such as capable, motivated, and organized attackers.

After selecting the attacker model, SPARTA's threat elicitation and risk analysis can be run. SPARTA will continuously perform pattern matching to find threats in the provided DFD model. For this it uses the VIATRA[2] query engine and graph-based pattern language. For each threat it finds, a risk analysis is conducted by sampling from the attacker, countermeasure strength, and DFD element value distributions. Using these samples, SPARTA combines them to calculate a risk estimate. This risk estimate, as well as the intermediate results (e.g., vulnerability) are included in the resulting threat list.

IV. Evaluation

We conducted an initial performance evaluation of SPARTA by running a threat and risk analysis on the WebRTC [27] reference architecture, shown in Figure 2, containing 42 DFD elements and resulting in 194 threats. The total analysis time (averaged over 10 runs) is 3.35 s, which includes loading the model, the query engine, pattern matching (for eliciting threats), risk analysis, and presenting the results to the user. Performing 100 runs for the just the threat elicitation and risk analysis (with 2000 samples per distribution) results in a mean value of 842 ms (95% CI: 718 ms–966 ms).

Additionally, we qualitatively evaluated the DFD solution enrichment of SPARTA, by comparing it with security property-based solutions such as the Microsoft Threat Modeling Tool 2016 [11], showing positive improvements in terms of semantic quality, traceability, separation of concerns, and dynamism [19].

Finally, we conducted an evaluation of SPARTA's risk-based threat prioritization on the open source whistleblower submission system SecureDrop [28], showing that the security countermeasures implemented in SecureDrop more often correspond to the high-risk threats identified by our tool than the low-risk threats [29].

V. Ongoing Work

We are currently extending SPARTA on three fronts. First, adding a security or privacy solution to a DFD model can be made more convenient by automatically instantiating any new elements and binding existing DFD-model elements to the security roles.

Second, the application of a specific security solution in an existing model can be extended to provide decision making and trade-off support to aid in the evaluation of multiple alternative security or privacy solutions, by evaluating their impact (i) on the DFD-model itself, (ii) on the resulting threat list, and (iii) on the calculated risks.

[2]https://projects.eclipse.org/projects/modeling.viatra

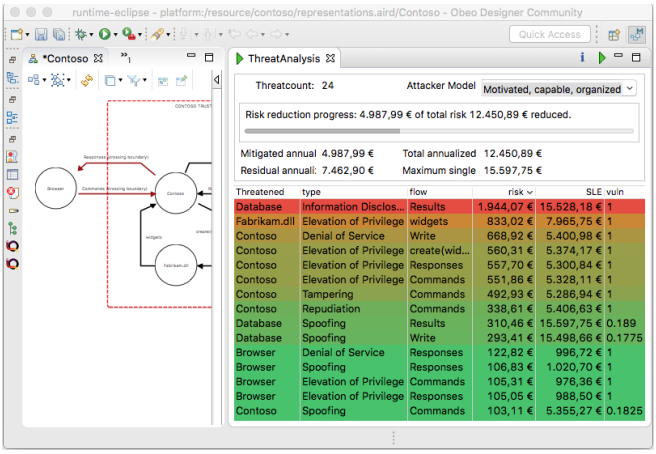

Fig. 3. Screenshot of SPARTA showing an example DFD model and the associated list of threats, color-coded based on the threat's calculated risk.

Finally, we also intend to centralize the security and privacy solution and threat knowledge bases of SPARTA, including the corresponding estimates of countermeasure strength, for example by fetching them from a web-service. This allows dynamic updates, keeping up-to-date with recent advancements in the field.

In the future, we intend to perform a user study on the approach realized in SPARTA, to assess the trade-off in effort between enriching the model with estimates and manually triaging the threats.

VI. Conclusion

Existing threat modeling tools lack an approach for prioritization that is grounded in data related to the security goals of the system under consideration. Additionally, risk analysis tools are disconnected from the concrete design of the system and the threats that such a design encompasses.

Our SPARTA tool addresses this disconnect by combining both DFD-based threat modeling, enriched with security and privacy solutions, and risk analysis simulations based on concrete element value estimates, countermeasure strengths, and attacker types.

By conducting a security analysis in this manner, a prioritized list of threats can be elicited, where the priority is based on actual estimated risks grounded in the model data, attacker type, and countermeasure data.

This type of tool support provides analysis results which correspond more closely to reality, where nothing is perfectly secure, but countermeasures do introduce a reduction of the risk that a certain threat manifests itself. Additionally, the risk-enriched threat list enables the threat modeler to monitor progress in reducing and managing the overall risk.

Looking forward, the combination of threat modeling, risk analysis, continuous monitoring to detect the appearance of new threats, and integration with an external threat and security solution catalog takes an important step forward towards realizing continuous, threat analysis and risk assessment for software systems.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven and the imec PRO-FLOW Research Project.

References

[1] M. Alshammari and A. Simpson, "Towards a Principled Approach for Engineering Privacy by Design," 2016.
[2] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016," *Official Journal of the European Union*, vol. 59, no. L 119, pp. 1–88, May 2016.
[3] M. Howard and S. Lipner, *The Security Development Lifecycle*. Microsoft Press, 2006.
[4] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat Modeling: Uncover Security Design Flaws Using The STRIDE Approach," *MSDN Magazine*, vol. 6, Nov 2006.
[5] K. Wuyts, "Privacy Threats in Software Architectures," Ph.D. dissertation, Jan 2015.
[6] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011.
[7] T. DeMarco, *Structured Analysis and System Specification*. Yourdon Press, 1979.
[8] M. S. Lund, B. Solhaug, and K. Stølen, *Model-driven risk analysis: the CORAS approach*. Springer Science & Business Media, 2010.
[9] J. Freund and J. Jones, *Measuring and managing information risk: a FAIR approach*. Butterworth-Heinemann, 2014.
[10] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Poster: Knowledge-enriched Security and Privacy Threat Modeling," in *2018 IEEE/ACM 40th International Conference on Software Engineering Companion (ICSE-C)*, 2018, p. (to appear).
[11] Microsoft Corporation, "Microsoft Threat Modeling Tool 2016," http://aka.ms/tmt2016, 2016.
[12] F. Swiderski and W. Snyder, *Threat modeling*. Microsoft Press, 2004.
[13] P. Torr, "Demystifying the threat modeling process," *IEEE Security & Privacy Magazine*, vol. 3, pp. 66–70, 2005.
[14] A. Shostack, "Experiences threat modeling at microsoft," in *Modeling Security Workshop. Dept. of Computing, Lancaster University, UK*, 2008.
[15] D. Dhillon, "Developer-Driven Threat Modeling: Lessons Learned in the Trenches," *IEEE Security Privacy*, vol. 9, no. 4, pp. 41–47, Jul 2011.
[16] Open Web Application Security Project, "OWASP Threat Dragon," https://www.owasp.org/index.php/OWASP_Threat_Dragon, 2018.
[17] T. Antignac, R. Scandariato, and G. Schneider, *A Privacy-Aware Conceptual Model for Handling Personal Data*. Cham: Springer, 2016, pp. 942–957.
[18] K. Tuma, R. Scandariato, M. Widman, and C. Sandberg, "Towards security threats that matter," in *3rd Workshop On The Security Of Industrial Control Systems & Of Cyber-Physical Systems (CyberICPS 2017)*, 2017.
[19] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Solution-aware Data Flow Diagrams for Security Threat Modelling," in *Proceedings of SAC2018: The 6th track on Software Architecture: Theory, Technology, and Applications (SA-TTA)*, 2018, pp. 1425–1432.
[20] T. Rauter, N. Kajtazovic, and C. Kreiner, "Asset-Centric Security Risk Assessment of Software Components," *2nd International Workshop on MILS: Architecture and Assurance for Secure Systems*, 2016.
[21] Continuum Security, "Irius Risk," https://community.iriusrisk.com, 2018.
[22] ThreatSpec, "ThreatSpec," https://threatspec.org/, 2017.
[23] S. Türpe, "The Trouble With Security Requirements," *25th IEEE International Requirements Engineering Conference*, 2017.
[24] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen, "A system of security patterns," 2006.
[25] A. Bedra, "Adaptive Threat Modeling, GOTO Conference Chicago," 2017. [Online]. Available: https://www.youtube.com/watch?v=YTtO_TGV2fU
[26] D. Vose, *Risk analysis: a quantitative guide*. John Wiley & Sons, 2008.
[27] Google, "WebRTC Project," https://webrtc.org/, March 2018.
[28] Freedom of the Press Foundation, "SecureDrop — The open-source whistleblower submission system," 2018. [Online]. Available: https://securedrop.org/
[29] L. Sion, K. Yskout, D. Van Landuyt, and W. Joosen, "Risk-based Design Security Analysis," in *Proceedings - 2018 IEEE/ACM First International Workshop on Security Awareness from Design to Deployment (SEAD)*, 2018, p. (to appear).